The 256-Processor X-Class Exemplar: A First Year's Experience

Gary M. Gutt

Jet Propulsion Laboratory, Mail Stop 126-147, California Institute of Technology, Pasadena, CA, USA 91109

Abstract

The Jet Propulsion Laboratory and the California Institute of Technology jointly acquired a 256-processor **X-Class Exemplar** in 1997. The system has been used by over 300 users for numerous applications in science and engineering. This paper will first describe the setup and organization of the system (including disk storage, libraries, memory allocations, network connections, queue structures, subcomplexes, and the tape storage system). Then it will cover unique features of the system and our solutions to problems encountered. Finally, it will give examples of some of the applications that have been run to date.

1 Introduction

The Jet Propulsion Laboratory (JPL) and the California Institute of Technology (Caltech) have been pioneers in the field of **massively parallel processors (MPP)** for the past 15 years. As part of our efforts to provide our research and engineering communities with advanced facilities for MPP, **we have acquired a 256-processor X-class Exemplar** computer **system from the High Performance Computing Division of Hewlett Packard**. This system is a departure from earlier distributed-memory systems available at JPL and Caltech (including a Cray T3D, the Intel Delta, and two Intel Paragons) in that it is a **cache-coherent, non-uniform-memory architecture (CC-NUMA) computer**. This shared-memory architecture allows for efficient programming using either a message-passing model or a global-shared-memory model of programming of MPPs. This system is also four times larger than any other system delivered by HP's High Performance Computing Division.

—Sponsoring the [illegible]

2 Acquisition of the Exemplar

In late 1996, JPL, in cooperation with the National Aeronautics and Space Administration (NASA) headquarters and other NASA centers, was developing plans to provide the next generation of supercomputing resources for its science and engineering communities. At about the same time, the US National Science Foundation (NSF) was soliciting proposals under its Partnerships for Advanced Computational Infrastructure program. One of the groups which submitted a successful proposal was the National Partnership for Advanced Computational Infrastructure (NPACI), which was led by the San Diego Supercomputer Center, and which included Caltech/JPL as one of five resource centers.

As a result of these plans and proposals, an agreement was announced between Caltech/JPL and Hewlett Packard's Convex division to implement a long-term plan for acquisition and support of a series of HP supercomputers. Under this agreement, Caltech/JPL is acquiring (with funding from NASA and NSF) a series of MPP supercomputers, beginning with a 256-processor X-class Exemplar, the largest system available from HP in 1997. In the future, Caltech/JPL will obtain one of the first MPP supercomputers based on the Merced microprocessor. Also, Caltech/JPL is collaborating with HP in upgrading and debugging the current SPP-UX operating system for use on the 256-processor system. Finally, Caltech/JPL will work with HP in the transition to HP-UX for all of its MPP computers.

The first system was delivered and installed at Caltech's Booth Computing Center in the form of four 64-processor computers during the first half of 1997. During the next few months, these were independently operated as two 64-processor systems and one 128-processor system. All of these were wired together to form one 256-processor supercomputer in October, 1997, which has been named "neptune".

3 Our Exemplar's Architecture and Capabilities

This X-Class Exemplar is based on the PA8000 microprocessor running at 180MHz internal clock speed. Due to its pipelined, superscalar architecture and dual arithmetic logic units, it is capable of a peak speed of

720 MFlops. Each processor has an associated 1 MB instruction cache and 1 MB direct-mapped data cache.

A node consists of a group of 16 processors connected to 4 GB of random access memory through an 8 × 8, nonblocking, crossbar switch. The 16 nodes in our system are connected through a two-dimensional, 4 × 4, bidirectional communications network called a Coherent Toroidal Interconnect (CTI) ring. A portion of the memory on each node is designated as global-shared memory and may be accessed by any processor in the system through the CTI ring. Each node has at least seven 9 GB hard disk drives operating through at least two controllers, for a total of 1.08 TB of storage. Each disk has a peak transfer speed of 10 MB per second, and each controller has a peak speed of 36 MB per second. Each node has at least one, four-way striped, `/tmp` file system, which has a peak speed of 40 MB per second. Each node also has one High Performance Parallel Interface (HiPPI), which is used primarily for communication with the archival tape storage system. There is one 100 Mb per second Ethernet port for the system.

The current operating system is SPP-UX 5.3, a UNIX-based operating system originally developed by Convex Corporation. SPP-UX has all of the features of UNIX along with extensions such as accounting and support for shared memory and the CTI ring. The available compilers include Fortran 77, Fortran 90, C, C++, and the Portland Group's High Performance Fortran (PGHPF). A variety of mathematics and communications libraries have been installed, including HP MLIB LAPACK; HP MLIB SCILIB; HP MLIB VECLIB; Math77/mathc90 (a large library of mathematical functions in Fortran and C for single processors, developed at JPL); the Parallel Virtual Machine (PVM) and Message Passing Interface (MPI) Libraries (including some functions from MPI-2); and math libraries for MPPs (the Parallel Iterative Methods and ScaLAPACK libraries).

The tools for software development presently are: CXdb (for debugging), CXpa (for performance analysis), CXtrace and XMPI (for debugging message-passing programs), and syspic (a very useful graphical tool for monitoring the activity and usage of various resources across all of the processors and nodes in the machine). One heavily-used commercial application which has been obtained for the Exemplar is Gaussian 98, a program for calculating chemical-bond characteristics from first-principles quantum mechanics.

In order to make the best around-the-clock use of the majority of the processors, a batch queue system is used. The queueing software is called the Load Sharing Facility (LSF) and is written by Platform Computing Corporation. The details of the operation of the batch queues are covered in the next section.

4 Structuring the Exemplar for Users

4.1 Subcomplexes

Under SPP-UX, the processors are assigned to groups called subcomplexes. Each job running in a subcomplex may only use the processors assigned to that subcomplex. This allows multiple jobs to run on the machine in separate subcomplexes without interference to each other, but limits the maximum number of processors available to a single job to the number of processors in the largest subcomplex. One subcomplex which is always present is called the System subcomplex. It is here that we run most user interactive jobs (*e.g.*, user logins, compilations, simple debugging) and Unix system daemons. We have found that assigning one node (16 processors) to the System subcomplex is sufficient for handling all of these jobs (as long as no one tries to run any production jobs here). In order to run large production jobs without interference from jobs in the System subcomplex, we have established a Batch subcomplex consisting of the remaining 240 processors. Access to the Batch subcomplex is only available through the batch queue system; this allows the large majority of the machine's resources to be used around-the-clock with good efficiency. Although the subcomplex concept has proven useful in managing the resources of a MPP system, it is only available under SPP-UX, not HP-UX.

4.2 Batch queues

Users of MPP batch queue systems have a number of (sometimes conflicting) needs:

1) Most users prefer to do program development during weekday hours. For interactive debugging and testing purposes, it is very desirable to give users access to a moderate number of processors (up to about 1/4 of the total number of processors in the Batch subcomplex) immediately upon demand. This requires that only a limited amount of production work be allowed during weekdays.

2) Some jobs run best on the maximum number of processors available (240). Such a job can only be started when all other running jobs have completed, leaving a completely free and clear Batch subcomplex.

To accomodate such jobs on an at least once-a-day basis, a queue is needed which can run at some time each night, after all small batch jobs have been drained down.

3) Some jobs require many hours to execute. In order to maintain a fair and equitable availability of system time to all users, it was decided to run long jobs (up to 12 hours) on weekends.

4) Several jobs from the queues can be run simultaneously, without interference, with proper scheduling.

Several different batch queues have been established (using the LSF system) to meet these user needs:

1) During weekdays (6am to 6pm Monday through Friday), users wishing to debug moderately large jobs (up to 64 processors) for modest periods of time (up to 30 minutes of wallclock time) are given a queue where they can generally get processors immediately on request. Also during weekdays, another queue allows the use of up to 240 processors (if available) for up to 10 minutes to enable a quick checkout of a very large job before it is submitted to one of the night or weekend queues.

2) At all times, a moderate amount of production work can be done in a queue which handles jobs requiring up to 64 processors for 3 hours.

3) During weekday nights and weekends, very large production jobs can be run in a queue which allows the use of up to 240 processors for 3 hours.

4) Jobs requiring very long times (up to 12 hours) are run on weekends in two queues allowing the use of up to 64 processors or 240 processors.

5) All queues allow the option of running a job interactively (*i.e.*, the standard input and standard output of a batch job may be connected to the terminal from which the job was submitted). This enables a user to run a job interactively on a large number of processors (up to 240), subject to processor availability.

4.3 Disk and tape storage

There are several competitors for disk space on a supercomputer. Each user would like to store some files on a semipermanent basis (*e.g.*, binary executables, configuration files, input files, and source code). Also, some users require a very large amount of disk storage on a temporary basis for writing checkpoint and output files. The compilers, libraries, operating system, and other system administration files occupy substantial space. Finally, a significant amount of disk storage for swap space is needed.

In the Caltech/JPL Exemplar, each user is given a maximum of 100 MB of "home directory" space for semipermanent file storage. Also, for temporary storage of very large data files, each of the 16 nodes has a minimum of 15 GB in `/tmp` directories, with a total of 423 GB of temporary storage. Files may remain for up to five days in this temporary storage, and are automatically "scrubbed" after that time. Without this automatic scrubber, we find that some users tend to "take over" the temporary disk storage space as their own. Finally, on each node, 9 GB of disk space is reserved for swap space.

Long term storage of large files and datasets is provided in the form of a High Performance Storage System (HPSS). This system uses a hierarchy of disks and tapes to give users easy, automatic access to their data, around-the-clock, without operator intervention. The HPSS consists of two IBM RS6000 servers with 76 GB of disk and an IBM 3494 tape robot which currently holds 2,200 10 GB (uncompressed) tapes for a total capacity of approximately 30 TB (compressed). Data transfers between the HPSS and the Exemplar are carried by a HiPPI network using either the File Transfer Protocol (FTP) or the HPSS Software Interface (HSI).

4.4 Documentation

Documentation of the aspects of "neptune" that are unique to this installation is maintained on the Caltech Center for Advanced Computing Research Web site at `http://www.cacr.caltech.edu/local_docs/exemplar/`. Users can find here instructions on obtaining an account, programming, compiling, executing jobs in batch, and storing data in the HPSS.

5 Problems Resolved

5.1 MPI startup time

Early in our use of the 256-processor machine, it was noted that when a large MPI job (*e.g.*, a 240-processor job) was starting, the entire system would be firmly committed to this process, to the extent that user logins and execution of all other commands and jobs would be delayed for approximately five minutes. This situation led many users to believe that the system had crashed whenever one of these large MPI jobs began. The HP MPI group has substantially parallelized the startup process and improved its efficiency in the past year to the point that today the startup of any size MPI job does not interfere with other processes on the system.

5.2 Shared memory jobs on many processors

When the system was first assembled as a 256-processor machine, it was discovered that shared-memory jobs requiring more than 176 processors could not be run. This problem was fixed in the spring of 1998 by a patch to the shared-memory software libraries.

5.3 Allocation of random access memory

At reboot, the 4 GB of RAM on each node must be allocated to four areas: global-shared memory, node-private memory, buffer cache, and CTI cache. User programs written with the shared-memory model will usually prefer large amounts of global-shared memory; whereas programs written with the message-passing model will prefer large amounts of node-private memory, with just enough global-shared memory to allow the communications libraries to function well. Through experience over the past year, we have found that it is best to allocate approximately 15% of RAM to global-shared memory, 75% of RAM to node-private memory, 7% to buffer cache, and 3% to CTI cache.

6 Current and Future Challenges

As with any cutting-edge technology, there are always new challenges to meet in advancing the state of the art.

6.1 Oversubscription

When a job is started on a MPP, it is important to maintain good load balance among the processors in order to achieve high parallel efficiency. This usually requires that each processor work exclusively on that job, without swapping to execute other jobs. When a processor is scheduled to work on two or more jobs concurrently, this situation is called oversubscription. Unfortunately, the job scheduling function within the current operating system cannot be ordered to avoid oversubscription when scheduling MPP jobs in the Batch subcomplex. The load imbalance resulting from oversubscription of even one processor can dramatically increase the execution time of a MPP job. We are currently collaborating with Hewlett Packard software engineers to find ways of preventing oversubscription in the Batch subcomplex.

6.2 Improved debugging capabilities

The current version of CXdb cannot be used to debug optimized code; it will only function with code compiled at optimization level +O0. However, programming with shared-memory directives in Fortran or C on the Exemplar requires the use of optimization level +O3. As a result, it is not possible to debug shared-memory-directive code with CXdb. We will be working with HP to identify a replacement debugger that can be used for both message-passing and shared-memory programs.

6.3 Implementation of MPI-2

As HP upgrades its MPI capability to encompass the MPI-2 standard, we have worked with the HP-MPI group under the leadership of Paco Romero to select the functions of greatest interest to be implemented first and have helped in their testing and evaluation. The functions of greatest interest to our user community have been the one-sided communication functions, which allow reading and writing to another process's memory without a matching call being executed within that process.

6.4 The transition to HP-UX

In order to provide its customers with a single operating system on all of its computers, HP will be transitioning the Exemplar series of supercomputers from SPP-UX to HP-UX. In the past, HP-UX was restricted to systems having no more than 16 processors. HP is currently preparing an upgrade of HP-UX to handle more processors; we will assist them in evaluating and debugging this upgrade on a new 32-processor V-class system at Caltech/JPL. This transition is expected to be straightforward from the users' point of view due to the many similarities between SPP-UX and HP-UX.

7 User Applications

The transition of users' programs and data from other MPP supercomputers to the Exemplar has gone well. Our users' experience has been that the HP compilers tend to do a better job of locating and identifying errors and potential problems in source code than other vendors' compilers. Here are two examples of user applications which have been developed for the Exemplar:

7.1 MATPAR: a parallel extension to MATLAB

MATLAB is a mathematical software system which is normally run on single-processor computers and workstations and is widely used at JPL. Many users of MATLAB have found a need to evaluate functions involving very large matrices (*e.g.*, 2000 × 2000) and have learned that this is unacceptably slow on a single-processor computer. Paul Springer of JPL's High Performance Computing Systems and Applications Group has written a set of extensions to MATLAB called MATPAR which will perform these compute-intensive functions on a MPP supercomputer. When a user in a MATLAB session on a workstation invokes one of these MATPAR functions, a PVM session is started on the MPP, the data is transferred to the MPP, the function is calculated, and the answer is returned to the workstation. Current parallelized functions include: matrix-matrix multiplication, QR factorization, LU factorization, frequency response calculations, and several other functions. A parallel, double-precision, matrix-matrix multiply of two 40,000 × 40,000 matrices on the Exemplar has achieved 84 GFlops on 256 processors.

7.2 Synthetic Forces Express: a distributed battlefield simulation

Synthetic Forces Express is a distributed, parallel implementation of the Modular Semi-Automated Forces (ModSAF) Distributed Interactive Simulation (DIS) software. DIS software (of which ModSAF is an example) allows for the simulation of a battlefield using a combination of high-fidelity computer-simulated vehicles (tanks, trucks, aircraft, etc.), interactions among those vehicles, a terrain database, and support for human-in-the-loop interactions. The goal of SF Express is to design and implement a scalable communications architecture for very large-scale distributed interactive simulations. SF Express has been developed at Caltech by Sharon Brunett, Dan Davis, Tom Gottschalk, and Paul Messina.

When SF Express is run on a MPP, most processors are used for running the individual vehicle simulation programs, some processors act as routers for communications between vehicle simulations, some processors act as servers which store the terrain database, and a few processors act as gateways for external communications. Thus, on the Exemplar, each simulation processor is handling approximately 150 vehicles, is transmitting 4–8 KB/s to its local router processor, and is receiving 1 MB/s from that router.

In SF Express, the battlefield simulation is conducted on a "computational grid" of many MPP supercomputers (a "metacomputer") to allow for the simulation of as many as 100,000 vehicles (see `http://www.cacr.caltech.edu/sfexpress/`). Communications between processors within one machine are handled by MPI, while communications between machines are handled with sockets and multicast IP techniques. In March, 1998, the Caltech/JPL Exemplar handled 21,951 vehicles within a 100,298 vehicle real-time simulation conducted over a metacomputer of thirteen computers with 1,366 processors. The Globus Metacomputing Toolkit (`http://www.globus.org`) was used for security and resource management in support of this effort.

8 Conclusion

We have described the capabilities and configuration (from a user's point of view) of the 256-processor X-Class Exemplar, which has been in operation at JPL/Caltech for the past year. Problems encountered and resolved as well as current and future challenges have been reviewed. Finally, several user applications have been presented.

Acknowledgements. I would like to thank the members of the JPL Supercomputing Support Group (Edith Huang, Zach Isaacs, Fred Krogh, Bob Mackey, and Cris Windoffer), the JPL supercomputing user community (Paul Springer), and the Caltech Center for Advanced Computing Research (Sharon Brunett) for their help in preparing this paper.

The United States of America National Aeronautics and Space Administration sponsors for this system are the Office of Space Sciences and the Earth and Space Sciences Project under NASA's High Performance Computing and Communications (HPCC) Program. The National Science Foundation also provided substantial funding for the system through Caltech's participation in the National Partnership for Advanced Computational Infrastructure.